CS 134: Networks	Section 10: Final Review
Prof. Yaron Singer	Spring 2017

1 Matching

1.1 Key Results

Theorem 1 (Matching Theorem)**.** *A bipartite graph (with $|U| = |V|$) has a perfect matching iff it does not contain a constricted set of vertices.*

Theorem 2 (Existence and Optimality of Market-Clearing Prices)**.** *For any set of buyer valuations, there exists a set of market-clearing prices that produces a socially optimal outcome.*

1.2 Algorithms

We know how to show whether or not a perfect matching exists in a bipartite graph, but how do we go about finding that perfect matching? We can use the idea of augmenting paths to build bigger and bigger matchings.

```
procedure FindPerfectMatch
    M ← empty match
    while true do
        v ← node unmatched in M
        p ← AltBFS(v)
        if p = ∅ then
            break
        end if
        M ← Enlarge(M, p)
    end while
    return M, p
end procedure
```

where AltBFS(v) returns an alternating path starting from node v and $\emptyset$ otherwise; Enlarge(M, p) expands M by removing the match edges in p from M and adding the non-match edges in p to M. The algorithm performs AltBFS() so long as there are unmatched nodes and breaks if it cannot find one. Recall that AltBFS() will fail if we are at a perfect matching or there is a constricted set. Then once the algorithm returns, we can check the path p to see if we found a constricted set. Aside: matching algorithms are actually useful; consider working on Datamatch!

We know that market-clearing prices exist for any set of valuations, but how do we go about finding them?

```
procedure FindMarketClearingPrices
    valuations v
    prices p ← 0
    while true do
        p ← p − min_i p_i
        G ← PreferredSellerGraph(p, v)
        if G has a perfect matching then
            break
        end if
        S, N(S) ← FindConstrictedSet(G)
        for all s ∈ N(S) do
            p_s ← p_s + 1
        end for
    end while
    return p
end procedure
```

where PreferredSellerGraph and FindConstrictedSet do the obvious things. We can use FindPerfectMatch to check if there is a perfect match in the induced prefered-seller graph, or find the constricted set. Note: market-clearing prices have practical use; consider trying to figure out how to divvy up different rooms within an apartment or suite with your roommates.

2 Final Review

2.1 Definitions

These questions will ask you to compute various concepts and run algorithms on given examples.

a. Suppose we have sellers a, b, c selling three distinct houses and buyers x, y, z with the following valuations:

buyer	a	b	c
x	3	6	4
y	2	8	1
z	1	2	3

What are the final market-clearing prices computed using FindMarketClearingPrices?

b. Draw a graph. Run the basic and scaled PageRank algorithms on it.

2.2 True/False

For each statement, indicate whether it is true or false. If you indicate it is true, nothing else is required. **If you indicate it is false, clearly describe a counterexample or other refutation.**

a. Increasing the strategy set of every player necessarily makes all players worse off in equilibrium. Braess' paradox is a special case of this general phenomenon.

b. Given a borrow-a-book game on a line with 5 nodes, there are exactly 4 equilibria.

c. In the voter model, assume node u has opinion 1 after k time steps. For all networks, there exists a node v such that if node v had a different initial opinion, then u would have had opinion 0 after k time steps.

d. Given a vector (of probabilities) of positions of a random walk and a vector of PageRank scores at some time t on a same network, the same matrix is used to update these two vectors for the next time step.

e. In a matching market, the set of market-clearing prices computed via the price-increment algorithm we presented in lecture maximizes the total revenue of the sellers (compared to any other allocation and set of payments the buyers might make).

Solution:

a. False; this is not true in general. Braess' paradox is indeed a special case where this is true.

b. True.

c. False.

d. True.

e. Given a set of prices, this is true because all sellers get payoff their price $p_i \geq 0$. Considering all possible price settings, it may not necessarily be true as there could be some set of prices that extract more value from the buyers.

2.3 Multiple Choice

This format of question actually won't appear on the exam, but the questions are still useful. Choose one answer for each part.

a. Consider the marketing game:

		Firm 2	
		Low-Priced (L)	Upscale (U)
Firm 1	Low-Priced (L)	5,1	6,4
	Upscale (U)	4,6	3,1

A. Firm 1's best response to L is U.
B. (L,L) is one of the Nash equilibria of the game.
C. L is strictly dominant strategy for Firm 1.
D. L is strictly dominant strategy for Firm 2.

b. Consider a random walk on an undirected, connected graph, which takes a step with equal probability to any of the neighbors of the node it is currently at, and also remains at the current node with a probability in $(0, 1)$.

A. This process will visit every node of the graph infinitely often, no matter where it starts.
B. If v and w are any two nodes, as $t \to \infty$ the probability of the walk being at v at time t and the probability of the walk being at w at time t converge to the same number.
C. The process has a stationary distribution, in which some entries are necessarily equal to 0.
D. A and C.

c. The tightest bound on the computational complexity of updating the scores of the hubs and authorities algorithm is:

A. $O(|V| + |E|)$.
B. $O(|V| \cdot \min_i d(i))$.
C. $O(|V|)$.
D. $O(|V| \cdot |E|)$.

Solution:

a. C

a. A

a. D. We find the complexity by noting that we must iterate over all the nodes twice (once for authorities, once for hubs). At each node, we must iterate over its neighborhood (in- and out-). B is not correct because it should read the $O(|V| \cdot \max_i d(i))$. The next tightest upper bound would be D.

2.4 Short Answer

Wally's Café. Wally's Cafe has live music every evening. The cafe has seating for only 15 people, so showing up for the evening is enjoyable only when at most 15 people do so. With more than 15 people in attendance, it becomes unpleasantly crowded, such that it would be prefereable to have stayed home. This week there are 30 people that can attend.

a. Transform the situation into a game in which every player has two strategies: "Go" and "Stay". Choose payoffs in a way that is consistent with the preferences described above.

b. Are there any dominated strategies? Explain.

c. Find all pure-strategy Nash equilibria.

Galton-Watson Branching Process. Consider a version of the branching process we studied. The first person who applies a certain filter (i.e. a modification) to his or her Facebook profile picture is r. Some of r's friends notice it and may apply the filter as well. Now, their friends notice the filter and can in turn change their profile pictures, etc. Assume that if any person applies the filter, then with probability .3 one of his friends applies the filter; with probability .3 two of his friends apply the filter; and with probability .4 none of his friends apply the filter.

a. What is the probability that r is the only one who ends up applying the filter?

b. What is the probability that the process stops with r's friends—i.e., nobody after the people who got "infected" directly by r apply the filter?

c. What is the branching factor (expected number of filter-adoptions directly resulting from a user's adoption) equal to?

d. Assume there is an infinite population of *potential* users. What is the probability that the spread of the filter eventually stops?

e. Find the expected number of users who eventually apply the filter.

Solution:

a. .4

b. $.3(.4) + .3(.4)^2$

c. $\mu = .4(0) + .3(1) + .3(2)$

d. Let $d = \Pr[\text{probability of process dying}]$. Then we can solve the following equation (derived similarly to the relevant problem set):

$$d = .4 + .3d + .3d^2$$

which implies that $d = 1$. This makes sense because the branching factor is less than one, so the process will die out almost surely.

e. 10; you can either sum an infinite geometric series or use a similar approach as above.

Epidemics. Suppose there is an outbreak of an epidemic in its early stages within a livestock population. Suppose you want to minimize the expected number of animals infected with a budget of \$2 million. If you spend x dollars controlling the extent to which animals come into contact with each other, then you expect each animal to come into contact with $40 - \frac{x}{200{,}000}$ others. If you spend y dollars introducing sanitization measures to reduce the probability of transmission, then you expect the probability an infected animal passes the disease to another animal contact to be $.04 - \frac{y}{100{,}000{,}000}$. How would you allocate your budget between x and y?

Solution: Implicitly we are modeling this epidemic as a branching process. We want to minimize the branching factor of the process, i.e. minimize $(40 - \frac{x}{200000})(.04 - \frac{y}{100000000})$ subject to $x + y \leq 2000000$. We will maximize our budget, so we can substitute $y = 2000000 - x$, so that our problem is min $\left(4 \times 10^6 - \frac{1}{2}x\right)\left(2 \times 10^6 + x\right)$ subject to $0 \leq x \leq 2 \times 10^6$, which comes out to putting all money into y.

Computational Complexity. Give a concise justification for your answers to the following questions:

a. Given a graph with n nodes and m edges, what is the computational complexity of finding the diameter of the graph?

b. What is the computational complexity of building a Kleinberg graph in terms of its parameters l, k and n?

c. In a game with two players with d actions each, what is the computational complexity of performing iterated elimination of strictly dominated strategies?

Solution:

a. In order to calculate the diameter, we need the shortest path between each pair of nodes. We can do this by performing a full BFS starting from each node. Since each BFS is $O(N + M)$ and we perform N of them, the complexity is $O(N^2 + NM)$. (Bonus: you can use Floyd-Warshall to find all shortest paths in $O(N^3)$ time, but this is equivalent because in the worst case $M = N^2$.

b. We consider each of the steps in building the Kleinberg graph in turn. First, we need to build the lattice, which takes n steps (just iterate through the nodes). Next, we need to connect each node to neighbors within k steps away, which forms a square of side length $k\sqrt{2}$ around the node. Overall then, this takes $O(k^2 n)$. Finally, we have to create l long-distance edges per node. If we assume we are given the probability of drawing another node to connect to, this takes $\frac{ln}{2}$ time, the number of edges formed, for an overall complexity of $O(n + k^2 n + ln)$. If we are not given the probabilities and need to compute them ourselves, then we need compute the lattice distance from each node to any other node. Since this is a lattice, we can just use differences of the coordinates and this takes $O(n)$ time per node, $O(n^2)$ overall, so our runtime is $O(n + k^2 n + ln + n^2)$.

c. In order to detect a dominated strategy, we need consider one player's actions, and iterate through the payoffs compared to the other player's actions. If a strategy is never the best response to another player's actions, it is dominated and we can remove it. If all of the other player's actions are being considered, then this takes d time. In the worst case, at each stage of iterated elimination we can only remove one action from a player's strategies, so we would have to take d, then $d-1$, then $d-2$, etc. steps to finally reach a Nash equilibrium, so this is $O(d^2)$.

In order to detect a dominated strategy, we may need to remove strategies for agents until each has one remaining, which means we have to check $2d$, $2d-1$, $2d-2$, $2d-3$, or $O(d^2)$ actions, using the summation formula $\frac{n(n+1)}{2}$. For each action, we need to compare that action with $O(d)$ of that player's remaining actions as well as $O(d)$ of the other player's actions. Therefore, putting the two steps together, we have that iterated elimination takes $O(n^4)$ time.